




**FOX** embedded computers   
*the canny swiss solution*

## Application Note 05

### Neuen Font einbetten mit Hilfe des eigerFont Editors

erstellt: 16. April 2008  
letzte Änderung: 5. Oktober 2011

**Christoph Angst**  
© S-TEC electronics AG, CH-6300 Zug  
eiger@s-tec.ch  
www.s-tec.ch  
www.eigergraphics.com



Dieses PDF-Dokument und die zugehörige eigerScript-Applikation MYFO sind Teil der eigerDemoCD. Sie können beides auch von der Downloadseite unserer Homepage [www.eigergraphics.com](http://www.eigergraphics.com) herunterladen.

---

## Inhalt

<b>1</b>	<b>Einleitung</b> .....	<b>3</b>
<b>2</b>	<b>eigerFont Editor</b> .....	<b>4</b>
2.1	Entwickler und Bezugsadresse .....	4
2.2	Systemvoraussetzungen .....	4
2.3	Installation .....	4
<b>3</b>	<b>Funktionen des eigerFont Editors</b> .....	<b>5</b>
3.1	Aufbau des Programmfensters .....	5
3.2	Zeichentabelle .....	5
3.3	Font Preview Tool .....	6
3.4	Glyph Workbench .....	6
3.4.1	Individuelle Zeichenbreite ändern .....	7
3.4.2	Zeichen editieren .....	8
3.5	Global Font Properties .....	8
<b>4</b>	<b>Eigenen Font in eigerScript Projekt integrieren</b> .....	<b>9</b>



# 1 Einleitung

Die Firmware des FOX embedded Computers ist mit einer verhältnismässig kleinen Anzahl Systemfonts ausgestattet (vgl. Tabelle 1). In der Application Note „Fonts und ASCII-Codes“ erhalten Sie einen Überblick über die Standard-Fonts von eigerScript.

Wollen Sie für Ihre eigerPanel-Anwendung weitere Fonts verwenden, so können Sie diese innert kürzester Zeit mit Hilfe des eigerFont Editors aufbereiten und in das eigerFont Format (\*.EFF) konvertieren. Es steht Ihnen hierfür die ganze Palette der Windows-Schriftarten zur Verfügung. Auch zugekaufte TrueType-Schriften können Sie in das eigerFont Format umwandeln. Der eigerFont Editor bietet Ihnen auch die Werkzeuge, mit denen Sie bestimmte Zeichen bearbeiten oder neue Zeichen entwerfen können.

In dieser Application Note lernen Sie die wichtigsten Funktionen des eigerFont Editors kennen. Mögen Sie es bequem, so können Sie sich auch anhand eines **Demo-Films** in die Welt des eigerFont Editors einführen lassen: <http://www.eigergraphics.com/Flash/EFF-Film1.htm>. Den Demo-Film im Macromedia Flash Format finden Sie auch auf Ihrer DemoCD (ab April 2008).

Kapitel 4 (ab S.9) nimmt Bezug auf die Beispiel-Anwendung MYFO. Darin wird gezeigt, wie Sie Ihren eigenen Font, den Sie mit dem eigerFont Editor vorbereitet und konvertiert haben, in Ihre Anwendung integrieren. MYFO finden Sie in der eigerDemoCD unter dem Verzeichnis Application Notes oder auf der Download-Seite von [www.eigergraphics.com](http://www.eigergraphics.com). Auf der Download-Seite ist MYFO zusammen mit dieser PDF-Datei in der ZIP-Datei „Neuen Font einbetten“ verpackt.

Sie können den Programmordner MYFO direkt auf die CompactFlash™ Card speichern und auf dem eigerPanel 57 ausführen. Achten Sie darauf, dass die Verzeichnis-Struktur stimmt (inkl. obligatorische Startdateien FOXLOGOLEGI und START.FOX). Eine genaue Anleitung dafür finden Sie im „eigerScript – Schnelleinstieg“ (ebenfalls auf der CD oder auf der Download-Seite von [www.eigergraphics.com](http://www.eigergraphics.com)).

**Tabelle 1:** Standard-Schriftarten in eigerScript

Konstanten-Name in eiger Script	Font und nähere Bezeichnung
Font_Arial_7	System 7, Mikroschrift
Font_Arial_8	System 8, Kleinschrift
Font_Arial_10n	Arial 10, Normalschrift
Font_Arial_12n	Arial 12, Normalschrift
Font_Arial_14n	Arial 14, Normalschrift
Font_Arial_16n	Arial 16, Normalschrift
Font_Arial_20n	Arial 20, Normalschrift
Font_Arial_24n	Arial 24, normal System Grossschrift
Font_System_9n	System 9, Normalschrift
Font_System_9bd	System 9 bold, Titelschrift fett
Font_System_18bd	System 18 bold, Titelschrift fett
Font_Courier_9n	Courier 9 bold, FixedSpace Schrift
Font_Tekton_bold_24	Tekton 24 bold, Spezialschrift FW
Font_DigitalNumbers_16	Spezialschrift Digitalanzeige nur Ziffern und Punkt.
Font_DigitalNumbers_24	Spezialschrift Digitalanzeige nur Ziffern und Punkt.
Font_DigitalNumbers_32	Spezialschrift Digitalanzeige nur Ziffern und Punkt.
Font_DigitalNumbers_48	Spezialschrift Digitalanzeige nur Ziffern und Punkt.
Font_DigitalNumbers_64	Spezialschrift Digitalanzeige nur Ziffern und Punkt.

## 2 eigerFont Editor

### 2.1 Entwickler und Bezugsadresse

Der eigerFont Editor wurde vom Ingenieurbüro WEITNAUER MESSTECHNIK in Näfels ([www.weitnauer-messtechnik.ch](http://www.weitnauer-messtechnik.ch)) für die S-TEC electronics AG entwickelt. Es ist das einzige kostenpflichtige Entwicklungstool zum eigerPanel. Die Lizenz können Sie bestellen bei:

S-TEC electronics AG  
Industriestrasse 49  
CH-6300 Zug  
+41 (0)41 754 50 10  
[www.eigergraphics.com](http://www.eigergraphics.com)  
[eiger@s-tec.ch](mailto:eiger@s-tec.ch)

### 2.2 Systemvoraussetzungen

1. Windows 2000, XP
2. **.NET** Framework (diese Softwareplattform von Microsoft ist bei neueren Windows-PCs bereits standardmässig installiert). Microsoft .Net Framework finden Sie auf der DemoCD und auf der Downloadseite von [www.eigergraphics.com](http://www.eigergraphics.com). Sie können es auch gratis herunterladen unter [www.microsoft.com/downloads](http://www.microsoft.com/downloads).

### 2.3 Installation

Der eigerFont Editor basiert auf dem .Net Framework und braucht sich deshalb nicht durch Installation in die Tiefen des Windows-Betriebssystems einzunisten. Das ganze Programm besteht aus einer einzigen EXE-Datei. Ein Doppelklick auf das Datei-Symbol genügt, um den eigerFont Editor zu starten. Dabei ist es einerlei, ob die Programmdatei „EigerFontEditor.exe“ auf der Festplatte oder auf einem externen Speichermedium abgelegt ist. Wir empfehlen jedoch, ausführbare .NET-Applikationen auf lokalen Laufwerken abzulegen. Sollen sie von Serverlaufwerken gestartet werden, ist eine Anpassung der Sicherheitseinstellungen des .NET-Frameworks erforderlich.

## 3 Funktionen des eigerFont Editors

### 3.1 Aufbau des Programmfensters

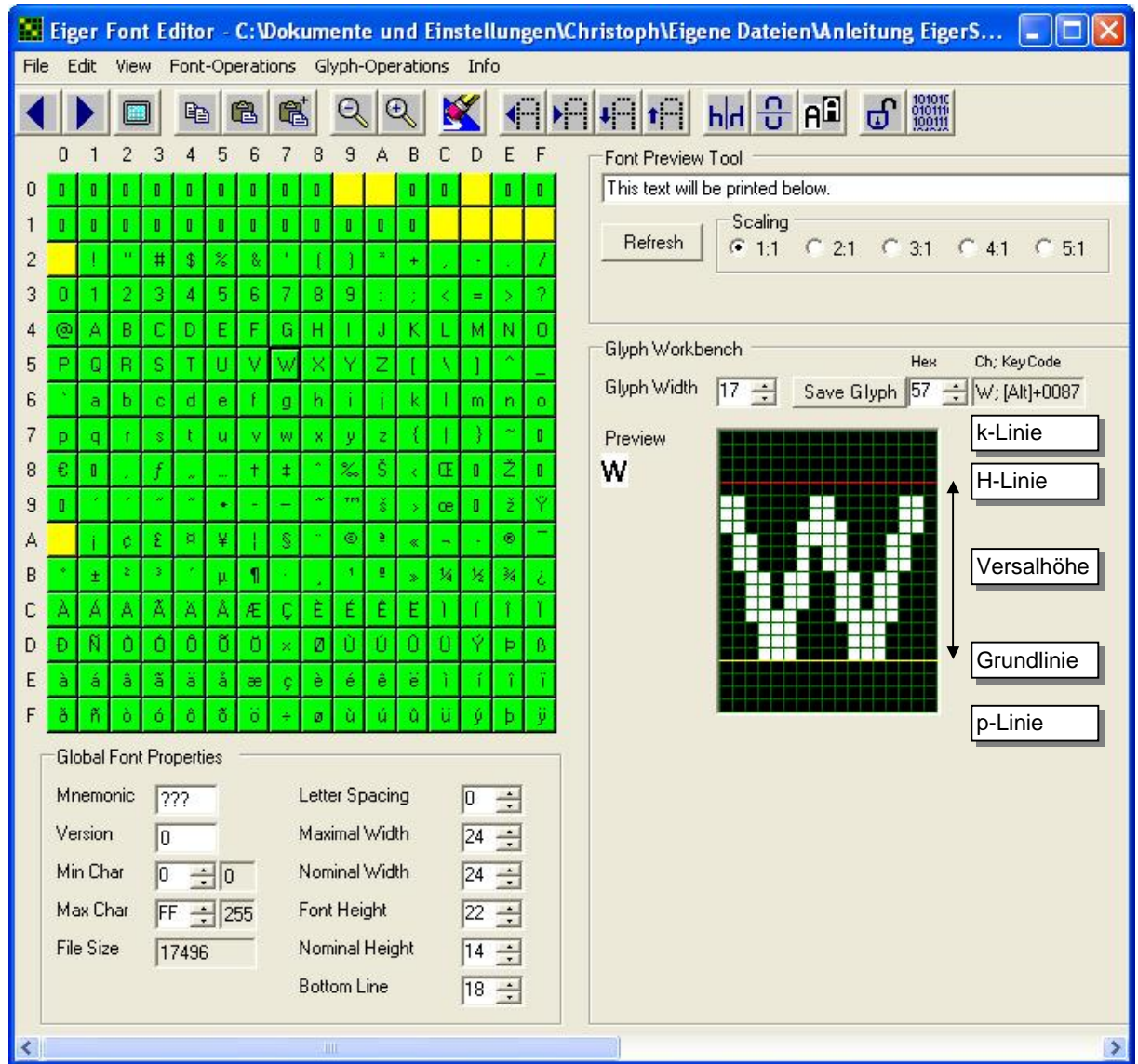


Abbildung 1: ScreenShot der Benutzeroberfläche des eigerFont Editors mit typographischen Bezeichnungen.

### 3.2 Zeichentabelle

In der linken Hälfte der Benutzeroberfläche des eigerFont Editors sind die maximal 256 Zeichen des importierten Fonts tabellarisch dargestellt. Die Tabelle umfasst den vollständigen ANSI-Zeichensatz. Ist die Tabelle **violett** gefärbt, dann ist noch kein Font geladen. Sobald ein Zeichensatz geladen wird, färbt sich die Zeichentabelle **grün**. Leere oder

gelöschte Zeichenplätze sind **gelb** dargestellt, wobei das offizielle Leerzeichen (0x20) bewusst leer ist. Zeichen, die Sie explizit nicht ins eigerFont Format übernehmen wollen, sind **rot** markiert (vgl. Kapitel 3.5, S.8).

Die Zeilen- und Spalten-Numerierung entspricht dem ASCII-Zeichencode. Demnach hat der Buchstabe "W" den ASCII-Code 57 hexadezimal bzw. 87 dezimal, in eigerScript als 0x57 geschrieben. Die verschiedenen Bezeichnungen finden Sie im Kopf der „Glyph Workbench“ (vgl. Abbildung 1 und Abbildung 4).

### 3.3 Font Preview Tool

Mit Hilfe des Font Preview Tools oberhalb der Glyph Workbench können Sie sich das Schriftbild anzeigen lassen. Was Sie ins Editierfenster schreiben, wird unterhalb im aktuellen Font dargestellt, sobald Sie auf **Refresh** drücken. Je nach Einstellung unter **Scaling** wird das Schriftbild in bis zu 5-facher Vergrößerung angezeigt. Dadurch sehen Sie z.B. schneller, welches Zeichen Sie zur Nachbearbeitung auf die Glyph Workbench „legen“ wollen.

Klicken Sie mit der rechten Maustaste auf ein Zeichen der Zeichentabelle, so wird es direkt ins Editierfenster des Font Preview Tools übertragen und im Schriftbild dargestellt.

### 3.4 Glyph Workbench

Per Mausklick auf ein Zeichenfeld bringen Sie das entsprechende Zeichen auf die Glyph Workbench in der rechten Hälfte des Fensters. In der Glyph Workbench können Sie die Darstellung bestehender Zeichen verändern oder auch ganz neue Zeichen gestalten. Das karierte „Zeichenbrett“ wird mit **zoom in** und **zoom out** vergrößert bzw. verkleinert. Hierzu dienen die beiden Buttons mit der Lupe oder die entsprechenden Schaltflächen unter Symbolleiste > View.

Der eigerFont Editor importiert und behandelt einen Font als Pixel-Font. Je grösser der beim Importieren gewählte Schriftgrad umso mehr Pixel braucht es, um ein Zeichen darzustellen. Dies kann sich markant auf den benötigten Speicherplatz auswirken.

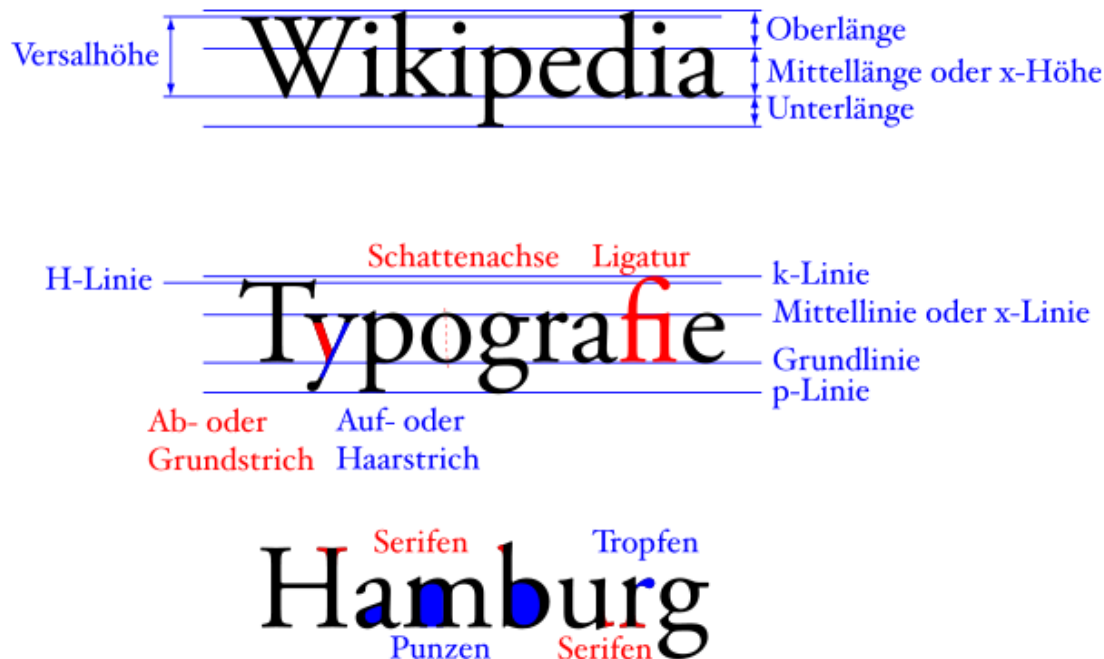
Geglättete Schriften sind i.d.R. eleganter als „gepixelte“ (vgl. Abbildung 2). Ein Pixel-Font wird nur durch zwei Farben dargestellt, z.B. schwarz/weiss. Dadurch entsteht ein Treppen-Effekt („Aliasing“). Für diese Darstellung spricht der geringe Bedarf an Speicherplatz, nämlich nur ein Bit pro Pixel. Treppenartige Kantenverläufe werden oft durch einen Algorithmus mittels Graustufen geglättet („Antialiasing“). Das eigerSystem unterstützt Antialiasing nicht. Soll die Schrift geglättet dargestellt werden, kann man sich mit Bildern behelfen, beispielsweise mit Screenshots von geglätteten Schriften.



Abbildung 2: Ziffern ohne und mit Antialiasing.

Die Höhe der Zeichen entspricht dem Schriftgrad, mit welchem der Zeichensatz eingelesen worden ist. In Abbildung 1 wurde beispielsweise der TrueType Font „Franklin Gothic

Medium“ im Schriftgrad 14 eingelesen, d.h. der Abstand zwischen der Grund- und der H-Linie entspricht 14 Pixeln (vgl. Abbildung 3).



**Abbildung 3:** Illustration einiger typografischer Begriffe (Kopiert aus [www.wikipedia.de](http://www.wikipedia.de) unter der Bedingung der GNU Free Documentation license. Autor: Brian Ammon).

### 3.4.1 Individuelle Zeichenbreite ändern

Im Kopf der Glyph Workbench finden Sie die wichtigsten Angaben zum angezeigten Zeichen (Abbildung 4). Die **Glyph Width** ist die Zeichenbreite inkl. individuellem Abstand zum nächstfolgenden Zeichen. Wenn Sie vom Buchstaben „W“ einen größeren Abstand halten, dann können Sie die Zeichenbreite von 17 Pixeln nach Wunsch verändern. Für den generellen Zeichenabstand ist das Feld „Letter Spacing“ zuständig (vgl. Kapitel 3.5).



**Abbildung 4:** Kopf der Glyph Workbench

Ihre Änderungen werden auf dem Zeichenbrett der Glyph Workbench unmittelbar ausgeführt. Zudem können Sie die Auswirkungen auch anhand der **Preview** überprüfen. Mit **Save Glyph** werden die Änderungen am betreffenden Zeichen gespeichert. Wechseln Sie ohne zu speichern zu einem anderen Zeichen, dann gehen alle am verlassenen Zeichen vorgenommenen Änderungen verloren.

Sie können Ihr verändertes Zeichen unter **Hex** einem anderen ASCII-Code (hexadezimal) zuweisen und diese Aktion mit Save Glyph speichern.

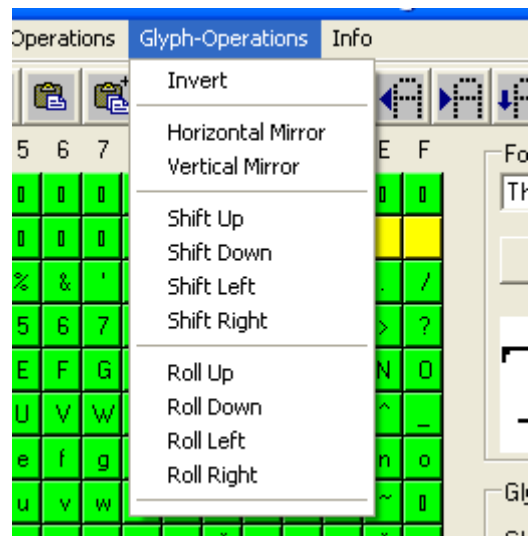


### 3.4.2 Zeichen editieren

Mit der **linken Maustaste** fügen Sie neue Pixel hinzu, mit der **rechten Maustaste** werden bestehende Zeichen-Pixel gelöscht.


In der Symbolleiste finden Sie unter **Glyph Operations** verschiedene Funktionen zum Editieren des Zeichens in der Glyph Workbench (Abbildung 5). **Invert** kehrt die Farben um – weisses Zeichen auf schwarzem Hintergrund und umgekehrt –, **Mirror** spiegelt das Zeichen horizontal bzw. vertikal und mit Shift verschieben Sie das Zeichen in Pixel-schritten in die gewünschte Richtung. Für diese Funktionen gibt es in der Symbolleiste auch entsprechende Buttons. Wenn Sie das Zeichen mit **Roll** verschieben, rückt das, was auf der einen Seite verschwindet, auf der andern Seite wieder ins Bild.

Mit **Copy** und **Paste** bzw. **Paste Add** (Symbolleiste > Edit oder den entsprechende Buttons auf der Symbolleiste) können Sie auch ganze Zeichen kopieren und damit ein anderes Zeichen ersetzen bzw. ergänzen. Wenn Sie beispielsweise den Gedankenstrich (0x97) kopieren und im TM-Zeichen (0x99) mit Paste Add einfügen, dann ist TM unterstrichen. Betätigen Sie darauf „Save Glyph“, dann ist der Unterstrich fester Bestandteil des TM-Zeichens geworden.



**Abbildung 5:** Glyph-Operationen zum Spiegeln, Verschieben oder Rollen eines einzelnen Zeichens.


## 3.5 Global Font Properties

Die Veränderungen in diesem Abteil betreffen immer den ganzen Font, also jedes einzelne Zeichen. Um unbedachte Änderungen in den Global Font Properties zu vermeiden, können Sie diesen Teil mit dem Button  sperren.

Auch die globalen Anpassungen können Sie mit dem Font Preview Tool testen.

Besonders nützlich sind die Font-Eigenschaften **Min Char** und **Max Char**. Hiermit können Sie den Zeichenumfang Ihres Fonts auf die Zeichen beschränken, die Sie wirklich benötigen. Genügen für Ihre Anwendung die Ziffern 0-9, so können Sie sich auf die Zeichen von „30“ (Min Char) bis „39“ (Max Char) beschränken. Höchstwahrscheinlich brauchen Sie aber auch noch das Leerzeichen (ASCII-Code 20), so dass Sie den Min Char vielleicht besser bei „20“ festlegen. Für viele Anwendungen kann auf die Steuerzeichen (00 bis 1F) und die internationalen Zeichen (z.B. 80 bis FF) verzichtet werden.

Damit können Sie vor allem bei grösseren Schriftfonts erheblich Speicherplatz einsparen. Eventuell sind sie bei grösseren Fonts sogar gezwungen sich zu beschränken, weil diese sonst die Speicherkapazitäten des Arbeitsspeichers eines eigerPanels schnell übersteigen.

Wenn Sie die entsprechenden Grenzwerte eingeben und diese mit ENTER bestätigen oder in der Symbolleiste den Button „Refresh“  drücken, werden in der Zeichentabelle alle ausgeklammerten Zeichen **rot** eingefärbt. Das Ausschlussverfahren machen Sie mit File > Save fest.

Im Feld **File Size** ändert sich die Anzahl Byte je nach Zeichenumfang.



Das Feld **Letter Spacing** bestimmt die Anzahl Pixel zwischen den einzelnen Zeichen. Bei der Einstellung „0“ besteht der Zeichenabstand allein aus der Anzahl freier Pixel, die individuell auf der rechten Seite des Zeichens angefügt sind.

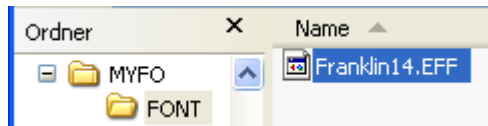
Die individuelle Glyph Width, einstellbar in der Glyph Workbench, kann die **Maximal Width** nicht überschreiten.

**Font Height** sollte nicht unbedacht verändert werden, denn damit wird bei zu kleiner Höhe der untere Teil jedes Zeichens gelöscht. Eine solche ungewollte Aktion können Sie nur mit Neuladen des Fonts „rückgängig“ machen.

## 4 Eigenen Font in eigerScript Projekt integrieren

Nachdem Sie Ihren Font mit Hilfe des eigerFont Editors bearbeitet und in das eigerFont Format (\*.EFF) konvertiert haben, können Sie diesen nun in Ihre Anwendung integrieren. Die Schritte für das Einbetten Ihres Fonts werden im Folgenden anhand des Projekts MYFO und der Font-Datei Franklin14.EFF erklärt.

1. Font im Verzeichnis Ihres eigerProjekts speichern, z.B. im Verzeichnis MYFO. Zugunsten der Übersichtlichkeit legen wir in MYFO ein Unterverzeichnis FONT an und speichern Franklin14.EFF dorthin (vg. Abbildung 6).



**Abbildung 6:** Speichern des eigenen Fonts im Projekt MYFO.

2. Font als Variable in die View einbinden (Variablendeklaration). Dies geschieht analog zur Deklaration eines String, Integer etc. (vgl. **A** in Beispiel-Code 1), z.B. `FONT MyFont = 'Franklin14.EFF'` (vgl. **A** in Beispiel-Code 1)
3. Font anwenden, z.B. in einer Subroutine. Der Befehl hierfür ist `Label.SetFont ( MyFont )`. Vgl. **B** in Beispiel-Code 1. Im Beispiel-Code 1 sehen Sie, dass die Variable MyFont nicht direkt in einen Style integriert werden kann. Im Style werden nur Konstanten akzeptiert.



**Bitte beachten** Sie, dass Sie zuerst eine Font-Variable deklarieren müssen, indem Sie ihr den korrekten Dateinamen als Zeichenkette in Hochkommata übergeben. Die direkte Anweisung `Label.SetFont('Franklin14.EFF')` darf NICHT verwendet werden! Der Compiler liest die Datei ein und erzeugt daraus die Fontdaten im Speicher. Anschliessend können Sie mit `Label.SetFont` den Font setzen, indem Sie der Methode den Variablennamen übergeben.

Die Fontdatei benötigt der Compiler – nicht das eigerPanel. Sie muss deshalb nicht unbedingt auf der CompactFlash Card vorhanden sein. Die Fontdaten werden direkt in das lauffähige EVI eingetragen.

**Beispiel-Code 1:** vollständiges Programm-Script des Beispiel-Projekts MYFO

```

;-----
; Titel      : Eigener Font   View 1
; File       : MYFO_001.EVS
;-----
; Compiler    : eigerScript
;
; System      : eigergraphics.com; FOXS embedded computer
;
; Beschreibung: Einbetten eines eigenen Fonts,
;              Beispiel zur Application Note "Neuen Font einbetten"
;
; Version     : Initialversion: 14.04.2008
;
; Autor       : Christoph Angst, S-TEC electronics AG
;
;-----
; (c) 2005-2008      S-TEC electronics AG, CH-6300 Zug; 041 754 50 10
;-----

EIGERPROJECT 'MYFO'      ; Projektbezeichnung: erster Teil des EVI-Dateinamens
EIGERVIEW 1             ; Viewnummer: zweiter Teil des EVI-Dateinamens: MYFO_001.EVI

IMPORT                  'DEF_eVM_OpCodes_0$60.h'           ; Token
IMPORT                  'DEF_eVM_Registers_0$60.h'         ; Register
FUNCLIB                 'DEF_eVM_Functions_0$60.lib'       ; Funktionsbibliothek

INCLUDEFILE             'DEF_eiger_Colors_0$60.INC'        ; Farbdefinitionen 12.03.06
INCLUDEFILE             'DEF_eiger_Types_0$60.INC'         ; eiger Definitionen

; Definitionen -----

STRING [60] Titel.$ = 'Eigener Font, MYFO_001.EVS: Franklin Gothic Medium 14'
STRING [60] Titel_Right.$ = 'erstellt: 14.04.2008'         ; Titelleiste rechts

; 1. Font einbinden (Variablendeklaration) =====
A FONT MyFont = 'FONT\\Franklin14.EFF' ; oder 'Franklin14.EFF' wenn direkt unter MYFO

; S U B R O U T I N E N -----

SUB Draw_Titel
; Titelleiste schreiben -----

Fill.LabelParameter ( Titel_Style )
Label.Text ( Titel.$ )

Fill.LabelParameter ( Titel_Right_Style )
Label.Text ( Titel_Right.$ )
ENDSUB

; 2. Font anwenden =====

SUB Write_Text
Fill.LabelParameter ( Text_Style )
Load.Geometry_XYWH ( 50 , 200 , 540 , 80 )
Label.SetFont ( MyFont )
Label.Text ( 'MyFont ist Franklin Gothic Medium, Schriftgrad 14' )
ENDSUB

```

```

; H A U P T P R O G R A M M _____
BEGINVIEW
    EVE.Init ( ) ; EVE ANNA initialisieren
    Display.ClearColor ( white )

    CallSubroutine( Draw_Titel )

    CallSubroutine ( Write_Text )

    LOOP
    ENDLLOOP
ENDVIEW

; S T Y L E S _____

SUB Styles

Titel_Style:
    INLINWORDS (0) ; entspricht eI.Pos_X1
    INLINWORDS (0) ; entspricht eI.Pos_Y1
    INLINWORDS (640) ; entspricht eI.Width
    INLINWORDS (18) ; entspricht eI.Height
    INLINWORDS (20) ; entspricht eI.SpaceLeft
    INLINWORDS (8) ; entspricht eI.SpaceRight
    INLINWORDS (0) ; entspricht eI.HorizontalAdjust
    INLINWORDS (0) ; entspricht eI.VericalAdjust
    INLINWORDS (black) ; entspricht eI.FillColor
    INLINWORDS (black) ; entspricht eI.BackColor
    INLINWORDS (black) ; entspricht eI.LineColor
    INLINWORDS (lightyellow) ; entspricht eI.TextColor
    INLINWORDS (Pos_left) ; entspricht eI.Position
    INLINWORDS (Orientation_0deg) ; entspricht eI.Orientation
    INLINWORDS (normal) ; entspricht eI.Appearance
    INLINWORDS (no_border) ; entspricht eI.BorderStyle
    INLINWORDS (Font_System_9bd) ; entspricht eI.FontNumber
    INLINWORDS (silver) ; entspricht eI.BackgroundColor

Titel_Right_Style:
    INLINWORDS (0) ; entspricht eI.Pos_X1
    INLINWORDS (0) ; entspricht eI.Pos_Y1
    INLINWORDS (640) ; entspricht eI.Width
    INLINWORDS (18) ; entspricht eI.Height
    INLINWORDS (20) ; entspricht eI.SpaceLeft
    INLINWORDS (8) ; entspricht eI.SpaceRight
    INLINWORDS (0) ; entspricht eI.HorizontalAdjust
    INLINWORDS (0) ; entspricht eI.VericalAdjust
    INLINWORDS (transparent) ; entspricht eI.FillColor
    INLINWORDS (black) ; entspricht eI.BackColor
    INLINWORDS (black) ; entspricht eI.LineColor
    INLINWORDS (lightsalmon) ; entspricht eI.TextColor
    INLINWORDS (Pos_right) ; entspricht eI.Position
    INLINWORDS (Orientation_0deg) ; entspricht eI.Orientation
    INLINWORDS (normal) ; entspricht eI.Appearance
    INLINWORDS (no_border) ; entspricht eI.BorderStyle
    INLINWORDS (Font_System_9bd) ; entspricht eI.FontNumber
    INLINWORDS (silver) ; entspricht eI.BackgroundColor

Text_Style:
    INLINWORDS (no_change) ; entspricht eI.Pos_X1
    INLINWORDS (no_change) ; entspricht eI.Pos_Y1
    INLINWORDS (no_change) ; entspricht eI.Width
    INLINWORDS (no_change) ; entspricht eI.Height
    INLINWORDS (20) ; entspricht eI.SpaceLeft
    INLINWORDS (8) ; entspricht eI.SpaceRight
    INLINWORDS (0) ; entspricht eI.HorizontalAdjust
    INLINWORDS (0) ; entspricht eI.VericalAdjust

```

C

```
INLINEWORDS (transparent) ; entspricht eI.FillColor
INLINEWORDS (as_FillColor) ; entspricht eI.BackColor
INLINEWORDS (as_FillColor) ; entspricht eI.LineColor
INLINEWORDS (blue) ; entspricht eI.TextColor
INLINEWORDS (Pos_center) ; entspricht eI.Position
INLINEWORDS (Orientation_0deg) ; entspricht eI.Orientation
INLINEWORDS (normal) ; entspricht eI.Appearance
INLINEWORDS (no_border) ; entspricht eI.BorderStyle
INLINEWORDS (Font_System_9bd) ; entspricht eI.FontNumber
INLINEWORDS (silver) ; entspricht eI.BackgroundColor
```

ENDSUB



Abbildung 7: ScreenShot der Anwendung MYFO.

